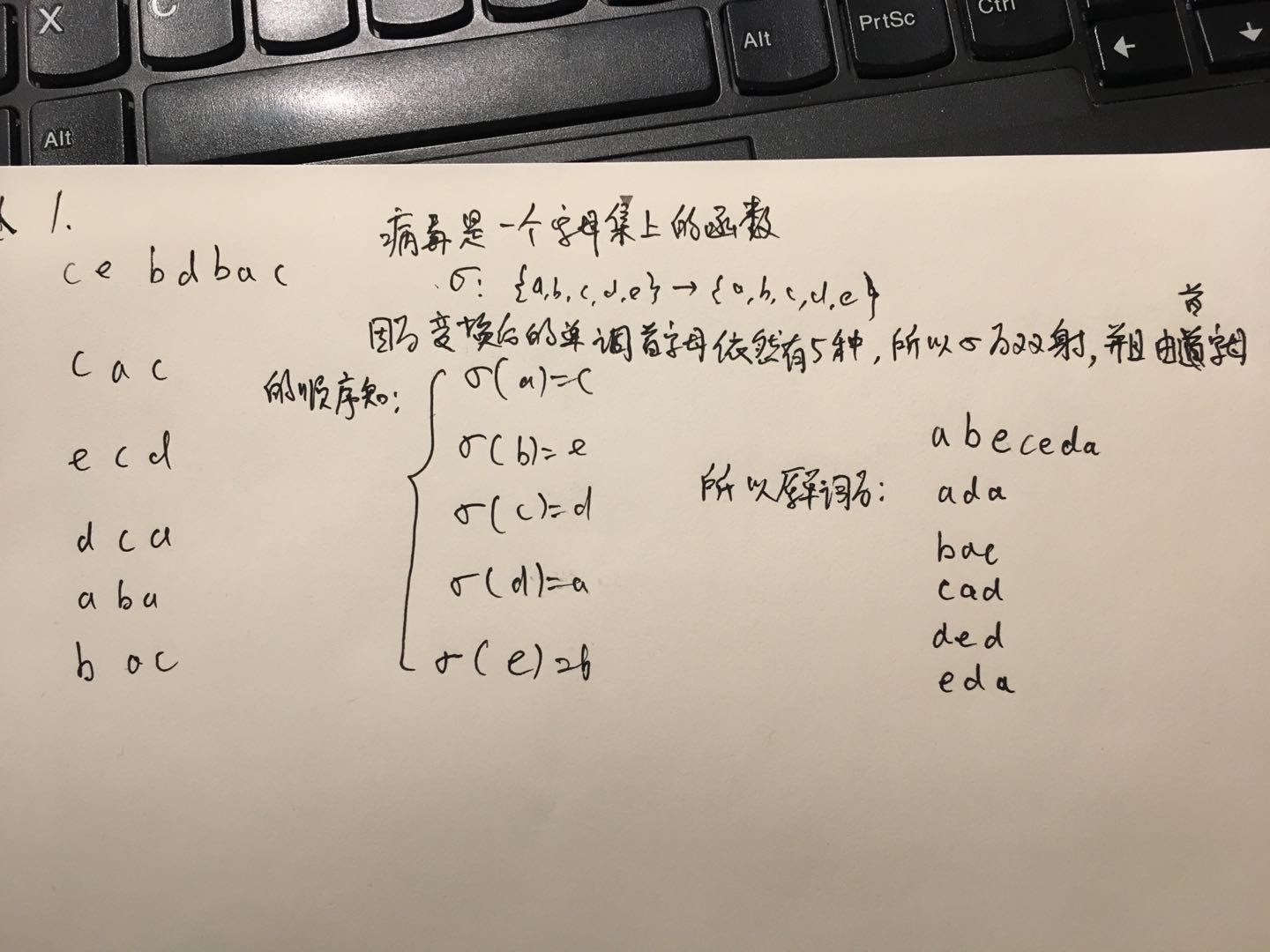
数算作业3

林涛 1600012773

1.



1. 这根本就不是一个算法。算法应该具有确定性，而“将集合V划分成两个集合V1和V2”这一步根本没有说明如何划分V1和V2。

3.

对于带排序数组a[1..n]，

1. 将数组分成L[1..m] 和 R[m+1..n]两部分，递归地对L和R排序.
2. 合并L和R.

合并L[1..m]和R[m+1..n]的方法是：

指针p=1, q=m+1

从1到n枚举指标i

从L[p]和R[q]中选出较小值,

如果是L[p], 则++p；

如果是R[q], 则将[p..q-1]平移到[p+1..q]这一段, 并把原来的R[q]放到p位置.

结束

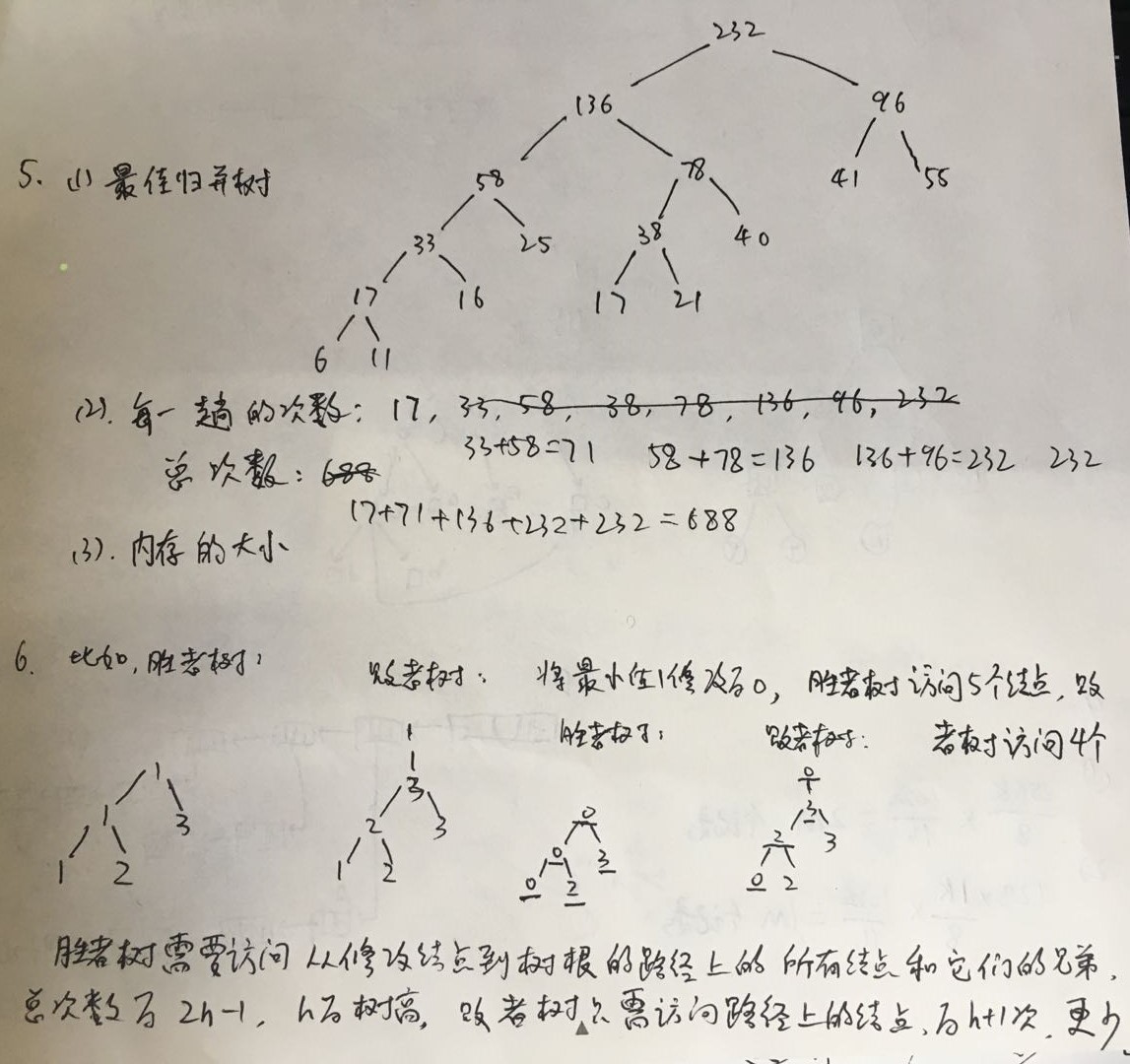
时间复杂度，对于长度为n的数组，合并过程中，最坏情况下，L中的元素比R中的都要大，则要进行 n/2 次整体的平移操作，每次移动L中的n/2个元素，共进行n^2/4次，所以时间复杂度递推式为：

f(n) = 2f(n/2) + n^2/4 = O(n^2)

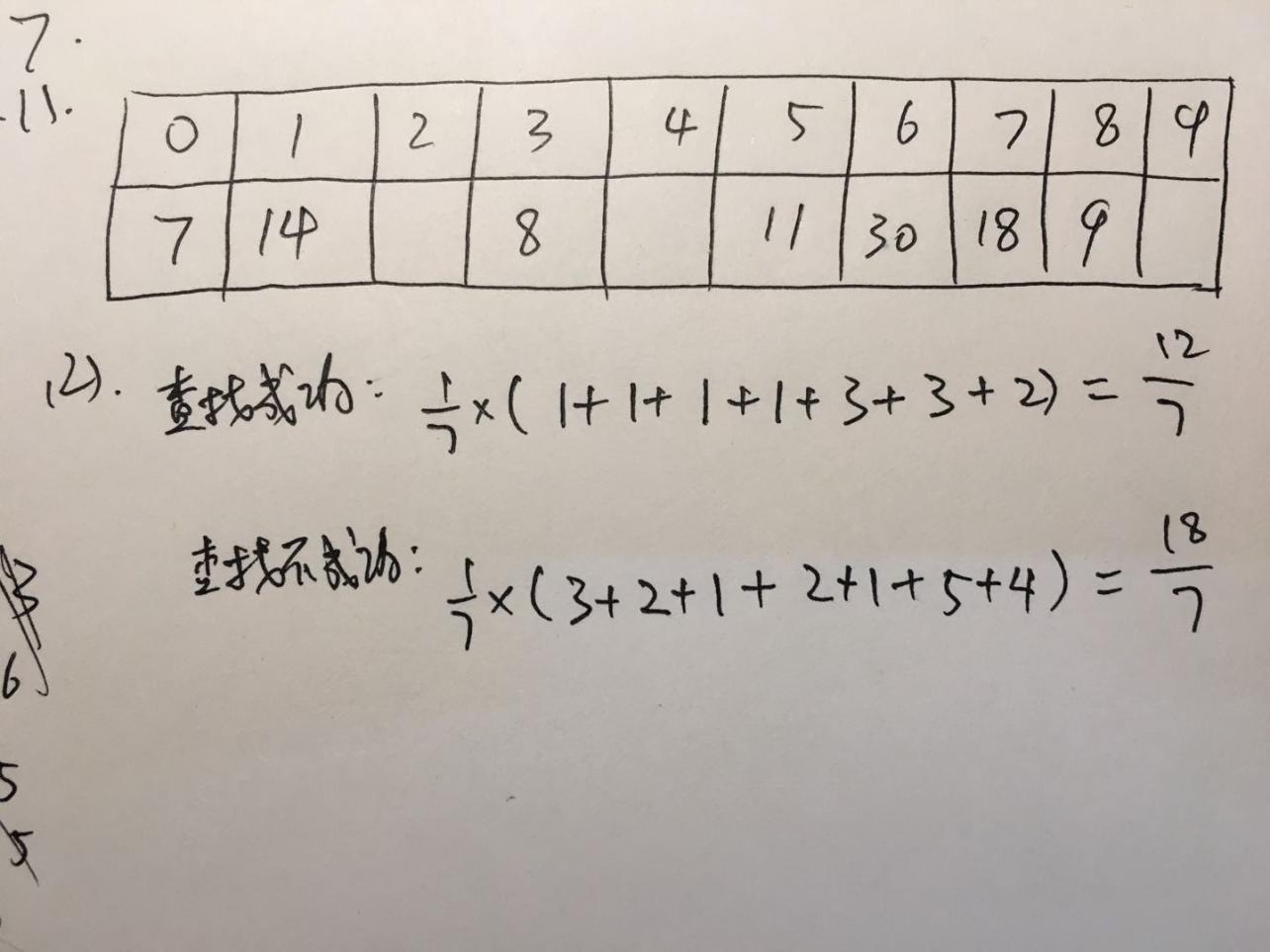
4.

1. 假设排序后的数组B[1..n]不是排好序的，则存在i<j，使得B[i]>B[j]，因为H具有单调性，所以H(B[i]) >= H(B[j])。 如果相等，那么B[i]和B[j]来自同一个桶，应该是排好序的，矛盾。所以是H(B[i]) > H(B[j])，但B[i]在B[j]前面，对应的桶却在B[j]的后面，与顺序扫描桶矛盾。因此B是排好序的。
2. 设第i个桶的元素个数为Si，总复杂度为:O(n)+ f(S0) + f(S1) + ... + f(Sm-1)
   1. 最好情况下，桶内已经有序，因此f(Si) = O(Si). 总：O(n)+O(S0) + O(S1) + ... = O(n)
   2. 最坏情况下，所有元素映射到一个桶并且逆序，插入排序的复杂度为O(n^2)， 因此总：O(n^2)
   3. 平均情况下，Si = n/m, 插入排序的平均复杂度为O(Si^2) = O(n^2/m^2), 总：O(n)+mO(n^2/m^2) = O(n^2/m)

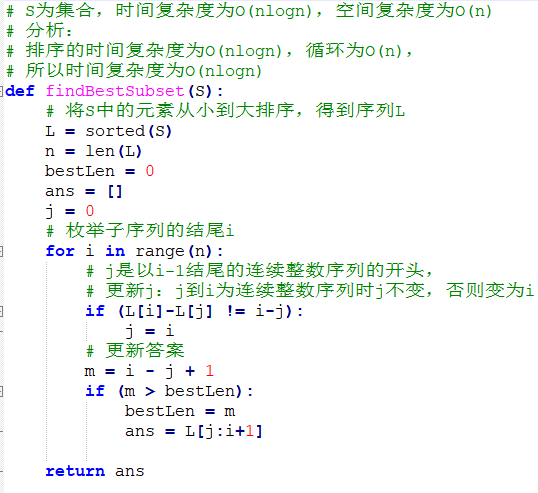
5，6.



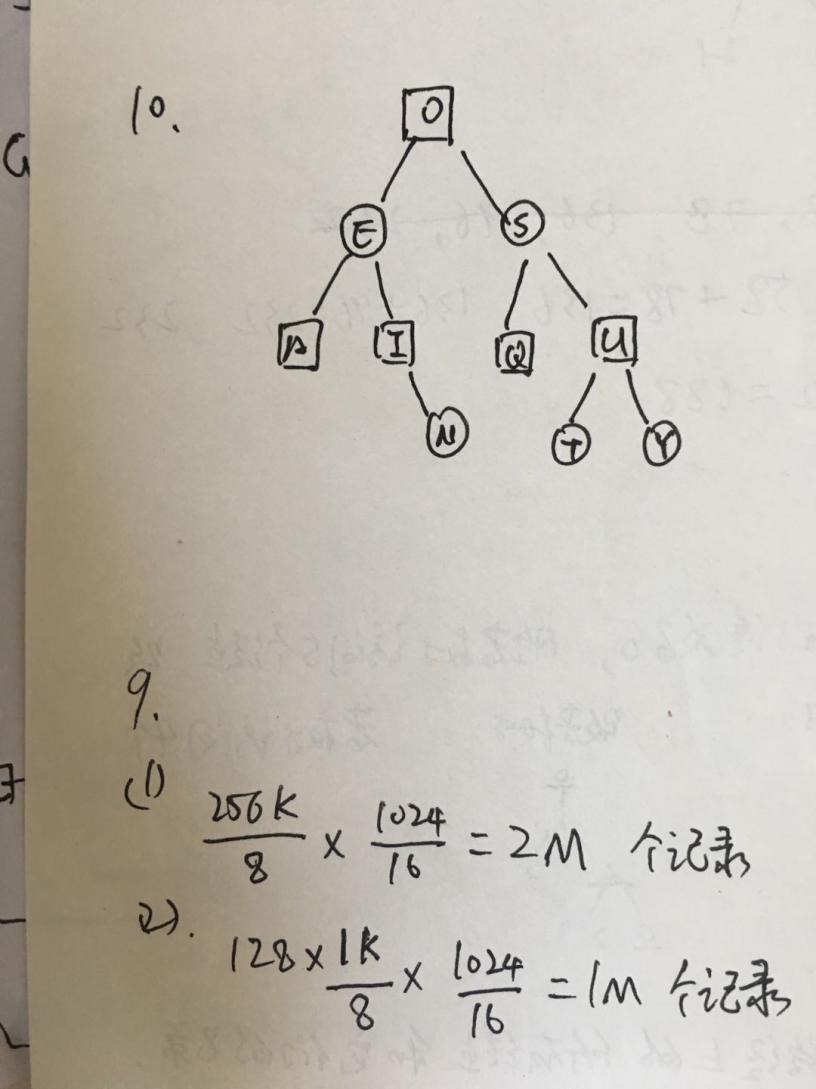
7.

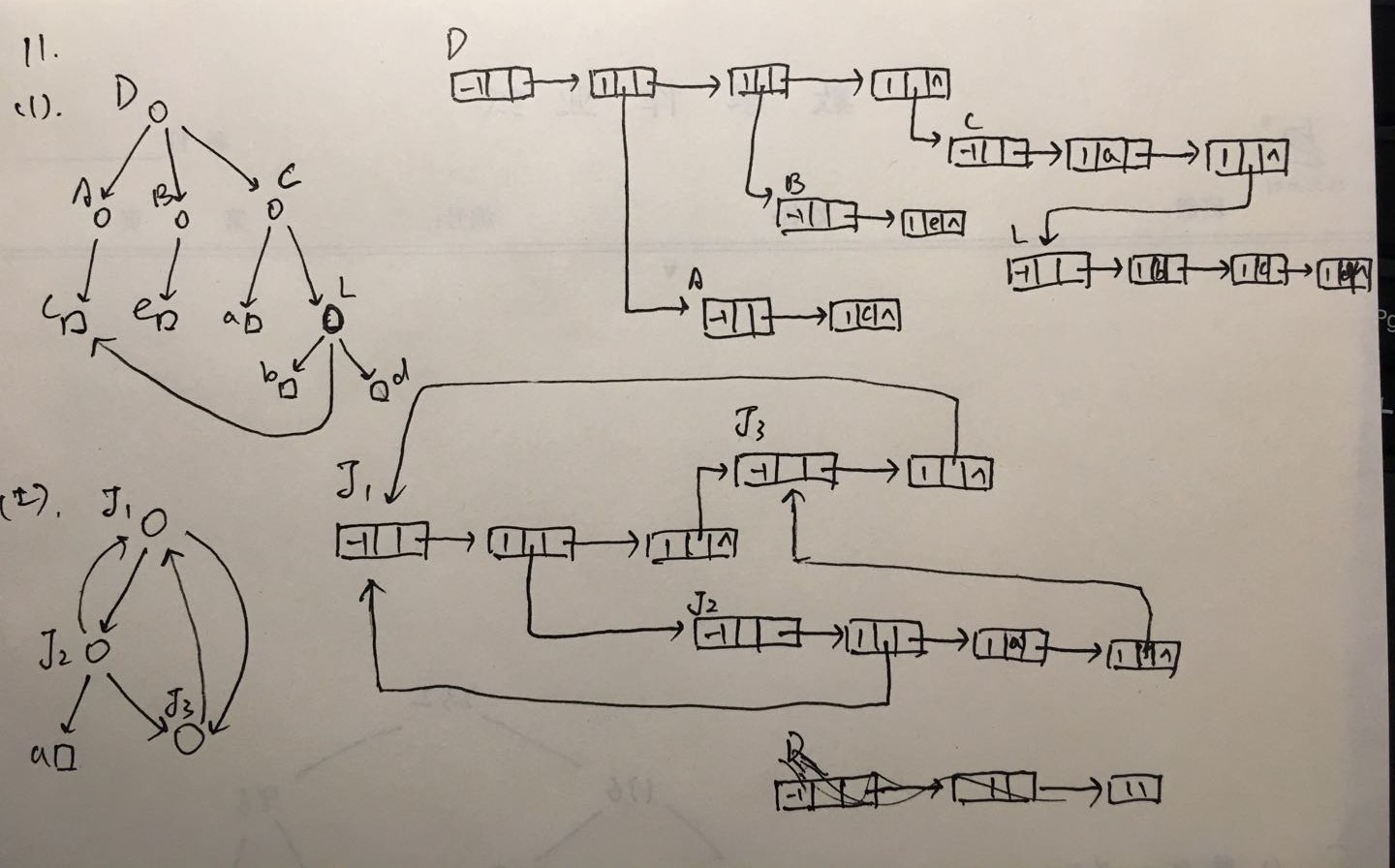


8.



9，10，11





12.

